고성능 Grid환경에서의 LDAP 시스템의 분산모델과 복제모델의 특성†

(Characteristic of Data Distribution and Data Replication based Model of LDAP System in High Performance Grid Environments)

권성호*, 김희철**
(Chenghao Quan, Hiecheol Kim)

요 약 최근, 국내외적으로 국가차원에서 Grid에 대한 다양한 연구가 이루어지고 Grid 컴퓨팅 기술에 대한 상용화의 추세가 보이면서 Grid 시스템에 참여하는 엔티티의 수가 날로 증가하고 있다. 따라서 Globus에 기반한 Grid환경에서의 기존의 LDAP시스템은 늘어나는 부하를 처리하기에는 충분하지 못하게 되었고 제공할 수 있는 성능의 한계를 초과하고 있다. 이를 위하여 고성능 Grid환경에 부합되는 새로운 LDAP시스템의 설계가 필수적이고 이를 위해서는 기존 LDAP시스템에 대한 성능분석이 우선시 되어야 한다. 하지만 기존의 대부분의 연구는 읽기 동작이 대부분을 차지하는 기존의 응용들을 위한 성능향상에 목적을 두고 있어서 쓰기 동작이 대부분을 차지하는 Grid환경에 적용하기에는 부적합하다. 본 논문에서는 노드수(n), 도착율(λ), 읽기확률(Pr)변화에 따른 분산 기반 모델과 복제 기반 모델에서 성능분석의 결과를 제시한다. 이를 위하여 M/M/1 큐잉모델을 기초로 기존 LDAP시스템에 대한 분산과 복제에 기반한 분석모델을 도출하고, 분석모델을 통한 성능분석의 결과를 제시한다. 본 논문의 목표는 이러한 성능분석의 결과를 바탕으로 고성능 Grid환경에 부합되는 새로운 LDAP시스템의 설계방향을 제시하고자 한다. 또한 이러한 결과들은 고성능 Grid환경에서 LDAP기반의 GIS시스템의 설계에 기초자료 활용할 수 있을 것으로 사료된다.

핵심주제어 : 디렉토리, 분산 모델, 복제 모델, 그리드

Abstract Recently, as the number of entities participating in the Grid system increased, the response time of LDAP system became inadequate. Consequently, we have to design new LDAP that suitable for high performance Grid environments. For this, researches about analysis of performance LDAP system are needed firstly. However, because researches are focused mostly on read operation optimized environments, so these result of researches are not directly applied to high performance Grid environments that write operation occupies most. In this paper, we provide overall results of analysis of performance of LDAP system with respect to number of node, query arrival rate, probability of read and so on. The analysis is based on an analytic performance model by applying the M/M/1 queuing model. Finally, based on the results, we suggest the direction for the design of high performance LDAP system and this research results can be applied as basic materials to design of GIS in high performance Grid environments as well as.

Key Words : Directory, Distribution Model, Replication Model, Grid

* Ph.D. Course, School of Computer and Communication Engineering, Daegu University.
** Professor, School of Computer and Communication Engineering, Daegu University.
† 본 연구는 2000년도 대구대학교 학술 연구비 지원에 의한 논문임.

1. 서 론

현재 기초과학과 산업기술 연구는 고속 연산, 대량의 데이터 처리, 첨단 장비의 공유 및 분산 협업 등이

필수적으로 수반된다. 이러한 요구를 만족시켜줄 수 있는 수단으로써 Grid는 지리적으로 분산된 고성능 컴퓨터, 대용량 DB 및 첨단 장비 등의 정보통신 자원을 고속 네트워크로 연동하여 상호 공유할 수 있도록 하는 컴퓨팅 방식이다. Grid는 바이오, 기상, 천문학, 인공지능 등의 첨단 분야의 기술 개발 뿐만 아니라 지구 생명의 규명, 실시간 항공기 설계 등을 가능하게 한다[1]. Grid프로젝트의 대표적인 예로 "SETI@home" (Search for Extraterrestrial Intelligence)은 인터넷에 연결된 컴퓨터를 이용하여 외계 지적 생명체를 찾는 거대한 프로젝트이다. 매일 전파망원경에서 흘러나오는 40Gbytes의 대량의 데이터는 전 세계의 3백여만 명 지원자의 컴퓨터로 전송되어 분석되어지고 있다. 지금까지 처리된 작업들을 한대 혹은 두 대의 슈퍼컴에서 처리한다면 수백만 달라 이상의 비용이 소모된다[2].

이러한 고성능 Grid환경을 구축하기 위해서는 Grid 내의 사용자, 관리자, 서비스, 하드웨어 등 다양한 자원에 대한 제반 정보서비스를 제공하는 Grid미들웨어 기술이 필수적으로 요구된다. GIS(Grid Information Service)는 기본적으로 자원 탐색(Discovery), 자원 할당(Scheduling and Allocation), 작업에 대한 모니터링(Monitoring)등 기능을 정의하고 구현하는 Grid 미들웨어 내의 핵심 소프트웨어 컴포넌트로써 지금까지 다양한 연구가 이루어지고 있다[2]. 이러한 연구의 일환으로서 Globus는 Grid미들웨어의 대표적인 연구 프로젝트로서 전 세계 관련 연구기관에서 Grid환경구축을 위한 프로젝트에 널리 활용하고 있다[3]. Globus 프로젝트에서 GIS 역할을 하고 있는 MDS (Metacomputing Directory Service)는 LDAP(Lightweight Directory Access Protocol)에 기반을 둔 계층적 구조를 가진 분산 디렉터리 시스템이다[4]. 이와 같이 Globus미들웨어에서의 GIS의 성능은 MDS를 구현하는 LDAP기반의 디렉터리 시스템의 성능에 좌우된다.

최근, 국내외적으로 국가차원에서 Grid에 대한 다양한 연구가 이루어지고 Grid컴퓨팅 기술에 대한 상용화의 추세가 보이면서 Grid 시스템에 참여하는 엔티티(사용자와 자원)의 수가 날로 증가하고 있다. 하지만 Globus에 기반한 Grid환경에서의 기존의 LDAP시스템은 늘어나는 부하를 처리하기에는 충분하지 못하게 되었고 제공할 수 있는 성능의 한계를 초과하고 있다. 또한 읽기 동작(Read Operation)에 최적화되어 있던 기존 LDAP시스템은 쓰기 동작(Write Operation)이 대부분을 차지하는 Grid환경에서는 부적합한 설계상의 교유한 결함도 가지고 있다. 따라서 고성능 Grid 환경에 부합되는 새로운 LDAP시스템의 설계가 필수적이고 이를 위해서는 기존 LDAP시스템에 대한 성능분석이 우선시 되어야 한다.

인터넷의 발전과 더불어 메타정보에 대한 효율적인 관리를 목표로 디렉터리시스템에 대한 관심이 증가하면서 지금까지 디렉터리시스템에 대한 다양한 연구결과가 발표되고 있다. YC. Jang.은 분산 디렉터리 시스템의 설계에서 복제메커니즘이 성능에 미치는 영향과 성능향상을 위한 적절한 복제비율을 그의 연구논문에서 제시하였다[5]. X. Wang은 LDAP시스템에 대한 전번적인 성능을 실험을 통해서 분석함으로써 성능향상을 위한 다양한 파라미터의 설정에 관한 결과를 그의 연구논문에서 제시하고 있다[6]. 하지만 이러한 논문들은 읽기 동작이 대부분을 차지하는 기존의 응용들을 위한 성능향상에 목적을 두고 있어서 쓰기 동작이 대부분을 차지하는 Grid환경에 적용하기에는 부적합하다. 최근의 W. Smith의 논문은 Grid환경에서의 LDAP시스템에 대한 성능분석의 결과를 제시하고 있다[7]. 하지만 이 논문의 주된 결과가 부하(Workload)에 대한 분석이란 점과 성능향상을 위한 몇 가지의 방향만 제시할 뿐 구체적인 방법을 제시하지 못하고 있다는 점이다.

본 논문에서는 노드수(n), 도착율(λ), 읽기확률(Pr)의 변화에 따른 분산 기반 모델과 복제 기반 모델에서 성능분석의 결과를 제시한다. 이를 위하여 M/M/1 큐잉모델을 기초로 기존 LDAP시스템에 대한 분산과 복제에 기반한 분석모델을 도출하고, 분석모델을 통한 성능분석의 결과를 제시한다. 이러한 성능분석의 결과를 바탕으로 고성능 Grid환경에 부합되는 새로운 LDAP시스템의 설계방향을 제시한다.

본 논문의 구성은 아래와 같다. 색션 2에서는 분석모델의 도출을 위한 LDAP 시스템과 분산 및 복제에 대한 기본적인 개념을 살펴보고 각각의 모델에서의 도착율(λ), 서비스율(μ) 및 응답시간(RT)을 유도한다. 색션 3에서는 도출된 분석모델을 기초로 성능분석의 결과를 제시하고 마지막으로 색션 4에서는 결론을 맺고 향후 연구방향을 제시한다.

2. LDAP 시스템 분석모델

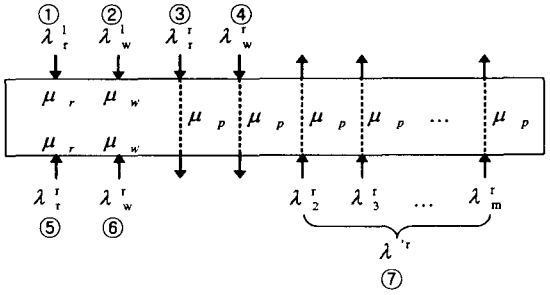
LDAP는 분산 디렉터리 서비스 프로토콜로서 클라이언트-서버 모델에 기반하고 TCP/IP 위에서 동작한다

[8]. 정보(Information)들은 DIB(Directory Information Base)에 DIT(Directory Information Tree)라고 불리우는 엔트리들의 계층적인 트리 모양의 구조로 저장되고 각각의 엔트리들은 DN(Distinguish Name)으로 구분되어진다. 분산 디렉터리시스템에서 정보들은 래퍼럴(Referral)을 이용하여 여러 개의 서버에 분산 저장되어진다. 래퍼럴은 포인터와 유사한 개념으로써 사용자가 요구하는 정보를 포함하고 있거나 혹은 알고 있다고 생각하는 서버의 주소(LDAP URL)를 가지고 있다. LDAP 동작(Operation)에는 검색(Search), 수정(Modify), 추가(Add), 삭제(Delete)등 여러 가지가 있지만 크게 엔트리에 대한 읽기 동작과 쓰기 동작으로 구분되어진다. 일반적인 목적으로 사용되는 LDAP시스템에서는 읽기 동작이 대부분을 차지하는 반면 Grid 환경에서는 쓰기 동작이 대부분을 차지한다.

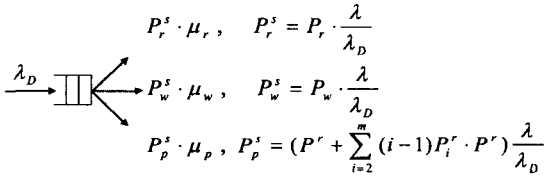
<표 1> 파라미터 설명

Parameter		Description
Scale	n	number of node
	m	number of referral
Arrival rate	λ	for single node
	λ^a	for system
	λ_i^r	for a request has i referrals
Service Rate	μ_r	for read
	μ_w	for write
	μ_p	for referral
	μ_{wb}	for write + broadcast
	μ_a	for acknowledge
Probability	P_i^r	for λ_i^r
	P^l	for local
	P^r	for remote
	P_r	for read
	P_w	for write
	P_r^l	for local read
	P_w^l	for local write
	P_r^r	for remote read
	P_w^r	for remote write

현재 LDAP시스템의 성능향상을 위한 다양한 기법들이 존재한다. 첫째, 디렉터리 정보들을 여러 개의 서버에 분산을 시키는 데이터 분산(Data Distribution)으로서 시스템을 구성하는 n개 노드들이 가지고 있는 디렉토리 정보간의 교집합이 없고 래퍼럴 개수 m 이 최대 값인 n-1을 가진다. 따라서 이러한 시스템은 분산의 정도에 따라 래퍼럴 개수가 늘어나면서 응답시간이 증가한다는 단점이 있는 반면 시스템이 처리할 수 있는 최대처리량이 증가한다는 장점이 있다. 둘째, 시스템을 구성하는 여러 개의 서버가 디렉터리정보들을 서로 복제하는 데이터 복제(Data Replication)로서 시스템을 구성하는 n개의 노드들이 디렉터리 정보를 완전 복제함으로서 래퍼럴 개수 m이 최소 값인 0을 가진다. 따라서 이러한 시스템들은 쓰기 동작에 따른 데이터의 변화를 시스템을 구성하는 다른 서버에 전달(Broadcast)해야 함으로써 그 오버헤드가 크다는 단점이 있는 반면 데이터의 복제에 따른 읽기 동작의 성능향상을 기대할 수 있다는 장점이 있다.



<그림 1> 분산 기반 모델



<그림 2> 분산 기반 모델의 서비스율

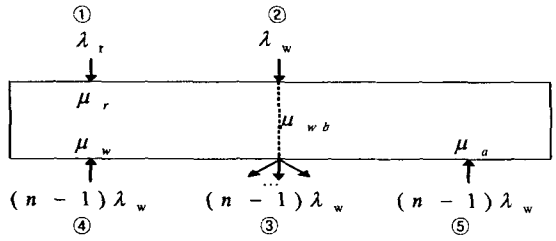
본 논문에서는 이러한 분산과 복제에 기반한 기존의 LDAP시스템에 대한 성능분석을 위하여 분산 디렉터리 시스템을 구성하는 각각의 노드들을 M/M/1시스템으로 간주하고 노드사이의 지연시간을 무시한다. 또한 최대 m개의 래퍼럴을 가지는 시스템에서 도착한 원격요구가 i개의 래퍼럴을 가질 확률은 $1/m$이라고 정의한다. 이와 같이 본 논문에서 제안한 분석모델은 아래와 같은 특징을 가지고 있다. 대부분 모델에서 래퍼럴을 1개로 가정한 것과 대조적으로 제안한 분산 기반 모델에서는 m개의 래퍼럴을 갖도록 설계하였다.

그리고 복제 기반 모델에서는 쓰기 동작이 다른 노드들로부터 쓰기 동작에 대한 응답을 받을 때까지 지연되는 Lock-Step Operation을 고려함으로써 실제의 분산 디렉터리시스템에 더 근접한 결과를 얻을 수 있도록 설계하였다. 표 1에는 본 분석모델에서 사용한 여러 가지의 파라미터에 대하여 설명하였다.

2.1 분산 기반 모델

그림1과 같이 분산 기반 모델은 로컬/원격 요구확률에 따라 도착한 요구가 로컬에서 서비스 될지 아니면 원격지로 래퍼럴 될지를 결정하게 된다. 제안하는 분석모델은 최대 m개의 래퍼럴을 가질 수 있도록 설계되었다. 여기서 m은 n-1이다. n개 노드로 구성된 분산 기반의 LDAP 시스템의 성능분석을 위하여 먼저 단일 노드에서의 요구 도착율(λ_D)과 서비스율(μ_D)을 구해야 한다.

그림 1과 같이 m 개의 래퍼럴을 갖는 분산 기반 모델에서 사용자로부터 도착한 요구율은 로컬 읽기 동작①, 로컬 쓰기 동작②, 원격(래퍼럴을 가지고 있음) 읽기 동작③, 원격 쓰기 동작④요구의 도착율의 합이다,



<그림 3> 복제 기반 모델

따라서:

$$\lambda = \lambda_r^l + \lambda_w^l + \lambda_r^r + \lambda_w^r \quad (1)$$

위의 식에 읽기/쓰기 확률과 로컬/원격 확률을 각각 대입하면 다음과 같이 나타낼 수 있다.

$$\lambda = (P_r^l + P_w^l + P_r^r + P_w^r) \cdot \lambda \quad (2)$$

위의 식에서 $(P_r^l + P_w^l + P_r^r + P_w^r) = 1$이다.

그 외에 사용자가 아닌 시스템을 구성하는 다른 노드로부터 도착한 요구율은 원격 읽기 동작⑤, 원격 쓰기 동작⑥, 해당하는 노드에 요구하는 정보가 없을 때 다른 노드로 래퍼럴 되는 동작⑦들의 요구의 도착율의 합으로 나타낼 수 있다, 여기서 동작⑦들의 요구의 도착율의 합은 아래와 같다.

$$\lambda'^r = 1 \cdot \lambda_2^r + 2 \cdot \lambda_3^r + \cdots + (m-1) \cdot \lambda_m^r,\ m = n-1,\ n \geq 3$$

$$= \sum_{i=2}^{m} (i-1)\lambda_i^r = \sum_{i=2}^{m} (i-1) P_i^r \cdot \lambda^r \quad (3)$$

따라서 하나의 노드에 도착하는 전체 도착율은 다음과 같다.

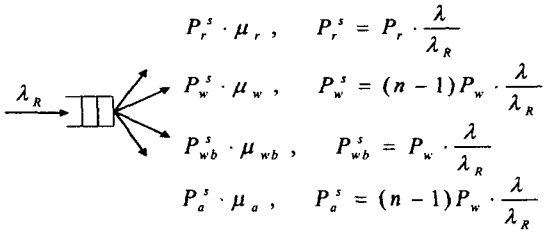
$$\lambda_D = \lambda_r^l + \lambda_w^l + \lambda_r^r + \lambda_w^r + \lambda_r^r + \lambda_w^r + \lambda'^r \quad (4)$$

위의 식에 식(2)와 식(3)을 대입하고 정리하면 아래와 같다.

$$\lambda_D = (1 + P^r + \sum_{i=2}^{m} (i-1) P_i^r \cdot P^r)\lambda \quad (5)$$

최대 m개의 래퍼럴을 가지는 시스템에서 도착한 원격요구가 i개의 래퍼럴을 가질 확률은 $1/m$로 정의되면서 위의 식에서 i개 래퍼럴을 갖는 원격요구의 확률은 $P_i^r = \frac{1}{m} P^r$과 같다. 따라서 식(5)는 다음과 같이 정리가 된다.

$$\lambda_D = (\frac{(m-1)}{2}(P^r)^2 + P^r + 1)\lambda \quad (6)$$



<그림 4> 복제 기반 모델의 서비스율

식(6)과 같은 도착율을 가지고 하나의 노드에 도착하는 요구들의 서비스율은 그림 2와 같이 나타낼 수 있다. 따라서 서비스율은 다음과 같다.

$$\mu_D = P_r^s \cdot \mu_r + P_w^s \cdot \mu_w + P_p^s \cdot \mu_p \quad (7)$$

위의 식을 다시 그림과 같이 도착율을 사용하여 표시하고 정리하면 아래와 같다.

$$\mu_D = (P_r \cdot \mu_r + P_w \cdot \mu_w + (\frac{(m-1)}{2}(P^r)^2 + P^r)\mu_p)\frac{\lambda}{\lambda_D} \quad (8)$$

M/M/1규잉모델에서 시스템의 큐잉 시간(평균 대기시간) T_D는 식(6)의 도착율과 식(8)의 서비스율에 의하여 아래와 같이 계산할 수 있다.

$$T_D = \frac{1}{\mu_D - \lambda_D} \quad (9)$$

따라서 로컬 요구에 대한 응답시간은 아래와 같이 계

산된다.

$$RT_D^l = \frac{1}{\mu_D - \lambda_D} \quad (10)$$

그리고 원격 요구에 대한 응답시간은 아래와 같이 계산된다.

$$RT_D^r = P_1^r \cdot 2T_D + P_2^r \cdot 3T_D + \cdots + P_m^r \cdot (m+1)T_D$$
$$= \sum_{i=1}^{m} P_i^r \cdot (i+1)T_D$$
$$= \frac{m+3}{2} P^r \cdot \frac{1}{\mu_D - \lambda_D} \quad (11)$$

하나의 노드에 도착한 요구의 평균 응답시간은 로컬 요구와 원격요구에 대한 응답시간의 평균으로써 나타낸다. 따라서

$$RT_D = P^l \cdot RT_D^l + P^r \cdot RT_D^r \quad (12)$$

위의 식에 식(10)과 식(11)을 대입하여 정리하면 아래와 같이 분산 기반 모델에서의 응답시간에 대한 식을 얻을 수 있다.

$$RT_D = \frac{P^l + \frac{m+3}{2} \cdot (P^r)^2}{\mu_D - \lambda_D} \quad (13)$$

2.2.복제 기반 모델

그림3과 같이 복제 기반 모델은 읽기/쓰기 요구확률에 따라 도착한 요구가 로컬에서 서비스 받은 후 시스템을 구성하는 다른 노드로 데이터의 변화를 전달해야 할지를 결정하게 된다. 제안하는 복제 기반 모델은 Lock-Step Operation을 지원하는 방식으로써 쓰기 동작은 다른 노드들로부터 쓰기 동작에 대한 응답을 받을 때까지 지연된다. n개 노드로 구성된 복제 기반의 LDAP 시스템의 성능분석을 위하여 먼저 단일 노드에서의 요구 도착율(λ_R)과 서비스율(μ_R)을 구해야 한다.

그림 3과 같이 사용자로부터 도착한 요구율은 읽기 동작①, 쓰기동작② 요구의 도착율의 합이다, 따라서:

$$\lambda = \lambda_r + \lambda_w = (P_r + P_w)\lambda \quad (14)$$

여기서 $(P_r + P_w) = 1$이다.

그 외에 사용자가 아닌 시스템을 구성하는 다른 노드로부터 도착한 요구율은 원격 읽기 동작④, 원격 쓰기 동작⑤ 요구의 도착율의 합으로 나타낼 수 있다

따라서 하나의 노드에 도착하는 전체 도착율은 다음과 같다.

$$\lambda_R = \lambda_r + \lambda_w + 2(n-1)\lambda_w, \quad n \geq 2 \quad (15)$$

위의 식을 다시 정리하면 아래와 같다.

$$\lambda_R = (1 + 2(n-1)P_w)\lambda \quad (16)$$

식(16)과 같은 도착율을 가지고 하나의 노드에 도착하는 요구들의 서비스율은 그림 4와 같이 나타낼 수 있다. 따라서 서비스율은 다음과 같다.

$$\mu_R = P_r^s \cdot \mu_r + P_w^s \cdot \mu_w + P_{wb}^s \cdot \mu_{wb} + P_a^s \cdot \mu_a \quad (17)$$

위의 식을 다시 그림과 같이 도착율을 사용하여 표시하고 정리하면 아래와 같다.

$$\mu_R = (P_r \cdot \mu_r + (n-1)P_w \cdot \mu_w + P_w \cdot \mu_{wb} + (n-1)P_w \cdot \mu_a)\frac{\lambda}{\lambda_R} \quad (18)$$

M/M/1규잉모델에서 시스템의 큐잉 시간(평균 대기시간) T_R는 식(16)의 도착율과 식(18)의 서비스율에 의하여 아래와 같이 계산할 수 있다.

$$T_R = \frac{1}{\mu_R - \lambda_R} \quad (19)$$

따라서 읽기 요구에 대한 응답시간은 아래와 같이 계산된다.

$$RT_{Rr} = \frac{1}{\mu_R - \lambda_R} \quad (20)$$

그리고 쓰기 요구에 대한 응답시간은 아래와 같이 계산된다.

$$RT_{Rw} = 3T_R = \frac{3}{\mu_R - \lambda_R} \quad (21)$$

하나의 노드에 도착한 요구의 평균 응답시간은 읽기 요구와 쓰기요구에 대한 응답시간의 평균으로써 나타낸다.

$$RT_R = P_r \cdot RT_{Rr} + P_w \cdot RT_{Rw} \quad (22)$$

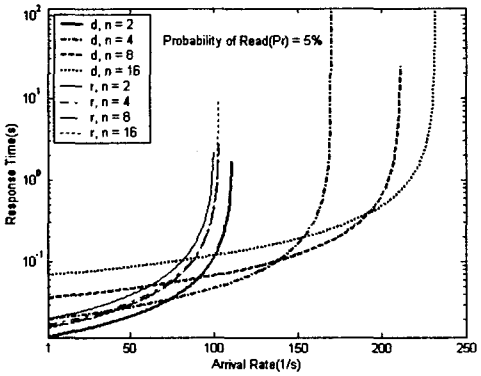
위의 식에 식(20)과 식(21)을 대입하여 정리하면 다음과 같은 복제 기반 모델에서의 응답시간에 대한 식을 얻을 수 있다.

$$RT_R = \frac{P_r + 3P_w}{\mu_R - \lambda_R} = \frac{1 + 2P_w}{\mu_R - \lambda_R} \quad (23)$$

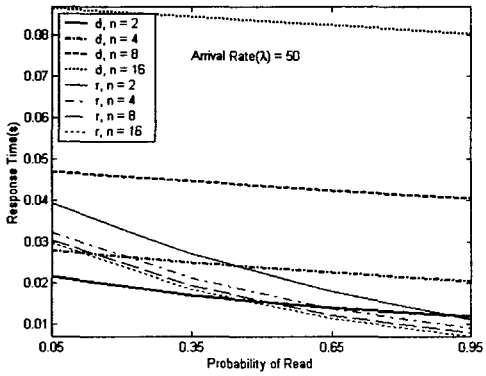
3. 성능분석

본 절에서는 앞에서 살펴본 분산과 복제에 기반한 분석모델을 토대로 노드수(n), 도착율(λ), 읽기확률(Pr)의 변화에 따른 분산 기반 모델과 복제 기반 모델에서의 응답시간의 변화를 다양한 측면에서 보여주었다. 실험에서 사용한 파라미터 값은 선행연구들의 결과를 기반

으로 한다. 이러한 파라미터들이 각각의 모델에서 응답시간에 미치는 영향을 알아봄으로써 각각의 모델에서의 파라미터에 따른 특성을 설명한다.

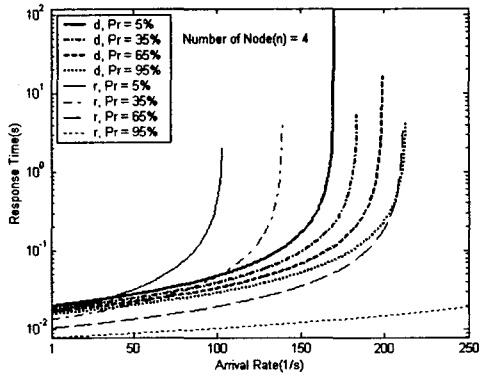


<그림 5> n, λr의 변화에 따른 분산 기반 모델과 복제 기반 모델에서의 성능 비교



<그림 6> n, Pr의 변화에 따른 분산 기반 모델과 복제 기반 모델에서의 성능 비교

그림 5는 읽기확률(Pr)이 5%일 때 분산 기반 모델과 복제 기반 모델에서의 도착율(λ)의 변화에 따른 응답시간의 변화를 노드수(n)에 따라 비교하여 나타내었다. 그림에서와 같이 분산 기반 모델에서는 n의 증가에 따라 응답시간과 최대처리량이 증가하는 것을 볼 수 있다. 이는 n의 증가에 따라 래퍼럴 개수가 증가하면서 응답시간이 늘어나기 때문이고 또한 n의 증가에 따라 시스템이 처리할 수 있는 용량이 늘어나면서 최대처리량도 증가하는 것으로 볼 수 있다. 이러한 최대처리량의 증가는 n의 증가에 따라 도착율이 250부근에서 포화가 되는 것을 볼 수 있다. 이는 포화최대처리량은 n의 값 보다는 실험에서 사용한 파라미터의 값들에 의하여 결정되기 때문이라고 판단된다. 이와 반대로 복제 기반 모델에서는 n의 증가가 응답시간이나 최대처리량의 변화에 거의 영향을 못 미친다는 것으로 알 수 있다. 이는 n의 증가에 따른 처리용량의 증가는 복제에 따른 과도한 오버헤드를 보상해주기 때문이고 또한 응답시간은 n의 증가와 무관하게 항상 로컬에서 서비스가 되기 때문에 거의 변화가 없는 것으로 판단된다. 따라서 복제 기반 모델에서의 응답시간과 최대처리량은 n의 변화보다는 실험에서 사용한 파라미터의 값들에 의하여 결정되는 경향이 있다고 생각되어진다.



<그림 7> λ, Pr의 변화에 따른 분산 기반 모델과 복제 기반 모델에서의 성능 비교

<표 2> 파라미터 값

Parameter	Value	Parameter	Value
μ_r	125	m	0 혹은 n-1
μ_w	60	P^l	(n-m)/n
μ_p	125	P^r	m/n
μ_{wb}	50	P_r(%)	5, 35, 65, 95
μ_a	330	P_w(%)	95, 65, 35, 5

이와 같이 분산 기반 모델에서 n의 증가에 따라 최대처리량이 증가한다는 장점이 있는 반면 응답시간도 증가한다는 단점도 있다. 복제 기반 모델은 n의 증가에 관계없이 꾸준히 응답시간을 유지한다는 장점이 있는 반면 최대처리량이 낮다는 단점도 있다.

그림 6은 도착율(λ)이 50일 때 분산 기반 모델과 복제 기반 모델에서의 읽기확률(Pr)과 노드수(n)의 변화에 따른 응답시간의 변화를 비교하여 나타내었다. 그림과 같이 읽기 확률(Pr)이 증가함에 따라 두 모델 다 응

답시간이 감소함을 볼 수 있었고 분산 기반 모델보다는 복제 기반 모델에서의 감소 폭이 더 크게 나타났다. 이는 μ_r이 μ_w보다 크면서 Pr의 증가에 따라 시스템의 전반적인 서비스율이 커지기 때문에 응답시간이 감소하고 그 외에 복제 기반 모델에서는 Pr의 증가에 따라 복제에 따른 오버헤드가 줄어들기 때문에 응답시간의 감소 폭이 더 큰 것으로 나타났다.

이와 같이 분산 기반 모델에서의 응답시간은 n의 값에 의하여 크게 좌우되고 복제 기반 모델에서의 응답시간은 Pr에 의하여 크게 좌우됨을 알 수 있다.

그림 7은 노드수(n)가 4일 때 분산 기반 모델과 복제 기반 모델에서의 도착율(λ)과 읽기 확률(Pr)의 변화에 따른 응답시간의 변화를 비교하여 나타내었다.

그림 7과 같이 읽기 확률(Pr)이 증가함에 따라 두 모델 다 응답시간과 최대처리량이 향상되는 것을 볼 수 있고 Pr이 65%이상일 경우 복제 기반 모델의 성능이 분산 기반 모델보다 우수한 것으로 나타났다. 복제 기반 모델에서 도착율이 150을 초과할 경우 Pr값이 65% 이상이어야 만 시스템이 포화가 되지 않고 정상적으로 동작할 수 있다.

이와 같이 읽기 확률(Pr)이 높을 경우 복제 기반 모델의 응답시간이 분산 기반 모델보다 우수한 걸로 나타나지만 복제 기반 모델의 문제점은 최대처리량이 상대적으로 작다는 것이다. 이러한 문제는 Pr의 증가에 따라 해소가 되지만 실제에 있어서 Pr은 우리가 제어할 수 있는 것이 아니고 응용에 따라 결정되는 경우가 대부분이다.

4. 결 론

본 논문은 기존의 LDAP시스템에 대한 성능분석을 위하여 분산과 복제에 기반한 분석모델을 도출하고, 분석모델을 통한 분산 기반 모델과 복제 기반 모델의 특성 분석의 결과를 제시한다.

분산 기반 모델은 n의 증가에 따라 최대처리량이 증가한다는 장점이 있는 반면 응답시간도 증가한다는 단점도 있다. 복제 기반 모델은 n의 증가에 관계없이 꾸준히 응답시간을 유지한다는 장점이 있는 반면 최대처리량이 낮다는 단점도 있다. 또한 분산 기반 모델에서의 응답시간은 n의 값에 의하여 크게 좌우되고 복제 기반 모델에서의 응답시간은 Pr에 의하여 크게 좌우됨을 알 수 있었다. 읽기 확률(Pr)이 높을 경우 복제 기반 모델의 응답시간이 분산 기반 모델보다 우수한 걸로 나타나지만 복제 기반 모델의 문제점은 최대처리량이 상대적으로 작다는 것이다. 이와 같이 분산 기반의 모델은 응답시간을 희생하는 대가로 처리량의 향상을 얻을 수 있고 복제 기반의 모델은 읽기 동작이 대부분을 차지하는 경우 우수한 성능을 보여주는 반면 쓰기 동작이 대부분을 차지하는 경우 성능향상이 거의 없음으로 나타났다. 이러한 실험결과들은 향후 고성능 Grid환경에 부합되는 새로운 LDAP시스템의 설계방향을 제시하고 나아가서 고성능 Grid환경에서의 LDAP기반의 GIS시스템의 설계에 기초자료 활용할 수 있을 것으로 사료된다.

향후, 완전복제나 완전분산이 아닌 분산과 복제의 복합모델에서 복제비율을 적절히 변화하여 래퍼럴의 개수를 줄여 응답시간을 최소화하고 노드수를 증가함으로써 최대처리량을 최대화하는 노드수(n)와 래퍼럴(m) 사이의 트레이드오프에 관한 연구가 이루어져야 할 것이다.

References

[1] I. Foster, C. Kesselman, S. Tuecke. The Anatomy of the Grid: Enabling scalable virtual organizations. *Intl. Journal of Supercomputing Applications.* (to appear) 2001.

[2] K. Czajkowski, S. Fitzgerald, I. Foster, C. Kesselman. Gird Information Service for Distributed Resource Sharing. *Proc. 10th IEEE International Symposium on High Performance Distributed Computer (HPDC-10) , IEEE Press.* 2001.

[3] The Globus Project homepage http://www.globus.org

[4] I. Foster and C. Kesselman. Globus: A Meatacomputing Infrastructure Toolkit. *International Journal of Supercomputing applications.* 11(2):115-128, 1997.

[5] YC. Jang, KS Kim, JS. Woo, SS. An. The Replication Mechanism Analysis of OSI(Open System Interconnection) Directory System. *J. ENG. SCI. & TECH.* Vol. 33, pp. 49~61.

[6] X. Wang, H. Schulzrine, D. Kandlur, D. Verma. Measurement and Analysis of LDAP Performance. *International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS' 2000),* Santa Clara, CA, pp. 156--165, Jun. 2000.

[7] W. Smith, A. Waheed, D. meyers, J. Yan. An Evaluation of Alternative Designs for a Grid

Information Service. *HPDC 2000, IEEE Press*: 185-192, 2000.
[8] H. Johner, L. Brown, FS. Hinner, W. Reis, J. Westman. Understanding LDAP *IBM*. June, 1998.

권 성 호 (Chenghao Quan)

1992년 (중)연변대학교 전자공학과 졸업
2001년 대구대학교 대학원 정보통신공학과 졸업(공학석사)
2001년 3월 ~ 현재 대구대학교 대학원 정보통신공학과 박사과정
(관심분야 : GRID, 컴퓨터 네트워크)

김 희 철 (Hiecheol Kim)

1983년 연세대학교 전자공학과 졸업 (공학사)
1991년 University of Southern California (Computer Eng. M.S.)
1996년 University of Southern California (Computer Eng. Ph.D.)
1983년 ~ 1988년 (주)삼성전자 주임연구원
1996년 ~ 1997년 (주)삼성SDS 수석연구원
1997년 ~ 현재 대구대학교 공과대학 정보통신공학부 교수
(관심분야 : GRID, 병렬처리, 컴퓨터구조, 컴파일러)